

Morfologia matemática binária

- Modificação do operador acerto e erro (hit-or-miss)
- Esqueletização
- Afinamento

- | | |
|--------------------------|--|
| - erosão de A por B: | $A \ominus B = \{z (B)_z \subseteq A\}$ |
| - dilatação de A por B: | $A \oplus B = \{z (B)_z \cap A \neq \emptyset\}$ |
| - abertura de A por B: | $A \circ B = (A \ominus B) \oplus B$ |
| - fechamento de A por B: | $A \bullet B = (A \oplus B) \ominus B$ |

Acerto e Erro (*Hit-or-miss*): $A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$

a) $(A \ominus B_1)$

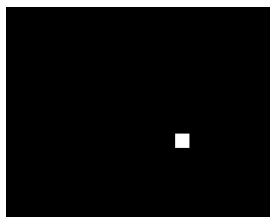
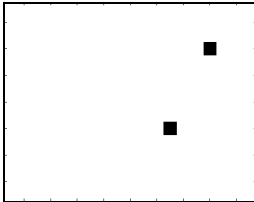
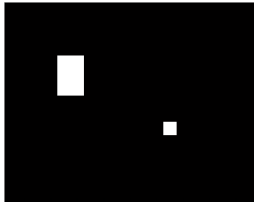
$$A \otimes B = (A \ominus B_1) \cap (A^\circ \ominus B_2)$$

```
d2 = floor(me/2);
d1 = ceil(me/2);
d4 = floor(ne/2);
d3 = ceil(ne/2);
S1 = zeros(m,n);
for i = d1:m-d2
for j = d3:n-d4
    ind = 1;
    for k = -d2:d2
    for l = -d4:d4
        if EE1(d1+k,d3+l) ~= 0
            a(ind) = EE1(d1+k,d3+l);
            b(ind) = I(i+k,j+l);
            ind = ind+1;
        end
    end
end
end
    if a == b
        S1(i,j) = 1;
    end
    clear a;
    clear b;
end
end
```

b) $(A^\circ \ominus B_2)$

$$A \otimes B = (A \ominus B_1) \cap (A^\circ \ominus B_2)$$

```
S2 = ones(m,n); % imagem de saída
for i = d1:m-d2
for j = d3:n-d4
    ind = 1;
    for k = -d2:d2
    for l = -d4:d4
        if EE2(d1+k,d3+l) ~= 0
            a(ind) = EE2(d1+k,d3+l);
            b(ind) = I2(i+k,j+l);
            ind = ind+1;
        end
    end
end
end
    if a == b
        S2(i,j) = 0;
    end
    clear a;
    clear b;
end
end
```



i = 10
j = 13

$$c) \quad A \otimes B = (A \ominus B_1) \cap (A \ominus B_2)$$

% intersecção entre S1 e S2:

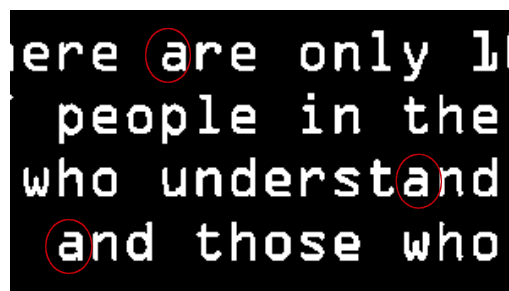
```
S = zeros(m,n);
for i = d1:m-d2
for j = d3:n-d4
    if S1(i,j) == 1 & S2(i,j) == 0
        S(i,j) = 1;
    end
end
end

[i j] = find(S == max(max(S)))
```

Exercício 2 da aula anterior:

Determine quantas vezes ocorre o padrão 'letra a' na imagem 'palavras.tif'.
Explique por que o operador *hit-or-miss* não reconheceu todas as ocorrências da letra a.

a



Acerto e Erro (Hit-or-miss):

$$A \otimes B = (A \ominus B_1) \cap (A \ominus B_2)$$

a) $(A \ominus B_1)$

```
d2 = floor(me/2);
d1 = ceil(me/2);
d4 = floor(ne/2);
d3 = ceil(ne/2);
S1 = zeros(m,n);
for i = d1:m-d2
    for j = d3:n-d4
        ind = 1;
        for k = -d2:d2
            for l = -d4:d4
                if EE1(d1+k,d3+l) ~= 0
                    a(ind) = EE1(d1+k,d3+l);
                    b(ind) = I(i+k,j+l);
                    ind = ind+1;
                end
            end
        end
        if a == b
            S1(i,j) = 1;
        end
        clear a;
        clear b;
    end
end
```



```
ind = me*ne;
S1 = zeros(m,n);
for i = d1:m-d2
    for j = d3:n-d4
        cont=0;
        for k = -d2:d2
            for l = -d4:d4
                if EE1(d1+k,d3+l) == I(i+k,j+l);
                    cont = cont+1;
                end
            end
        end
        if (cont/(ind)) >= 0.95
            S1(i,j) = 1;
        end
    end
end
```

Acerto e Erro (Hit-or-miss): $A \otimes B = (A \ominus B_1) \cap (A \ominus B_2)$

b) $(A \ominus B_2)$

S2 = ones(m,n); % imagem de saída

```
for i = d1:m-d2
    for j = d3:n-d4
        ind = 1;
        for k = -d2:d2
            for l = -d4:d4
                if EE2(d1+k,d3+l) ~= 0
                    a(ind) = EE2(d1+k,d3+l);
                    b(ind) = I2(i+k,j+l);
                    ind = ind+1;
                end
            end
        end
        if a == b
            S2(i,j) = 0;
        end
        clear a;
        clear b;
    end
end
```



```
ind = me*ne;
for i = d1:m-d2
    for j = d3:n-d4
        cont=0;
        for k = -d2:d2
            for l = -d4:d4
                if EE2(d1+k,d3+l) == I2(i+k,j+l);
                    cont = cont+1;
                end
            end
        end
        if (cont/(ind)) >= 0.95
            S2(i,j) = 0;
        end
    end
end
```

Acerto e Erro (*Hit-or-miss*): $A \otimes B = (A \ominus B_1) \cap (A \ominus B_2)$

c) $A \otimes B = (A \ominus B_1) \cap (A \ominus B_2)$

% intersecção entre S1 e S2:

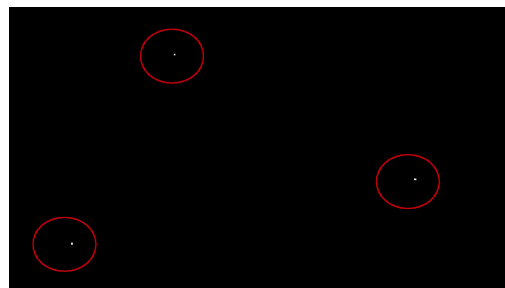
```
S = zeros(m,n);  
for i = d1:m-d2  
for j = d3:n-d4  
if S1(i,j) == 1 & S2(i,j) == 0  
S(i,j) = 1;  
end  
end  
end  
[i j] = find(S == max(max(S)))
```

Exercício 2 da aula anterior:

Determine quantas vezes ocorre o padrão 'letra a' na imagem 'palavras.tif'.
Explique por que o operador *hit-or-miss* não reconheceu todas as ocorrências da letra a.

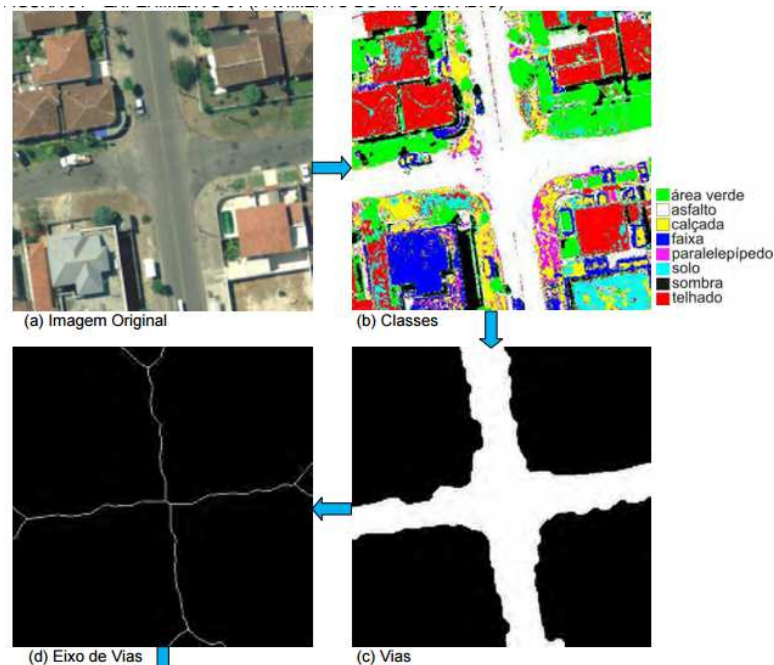
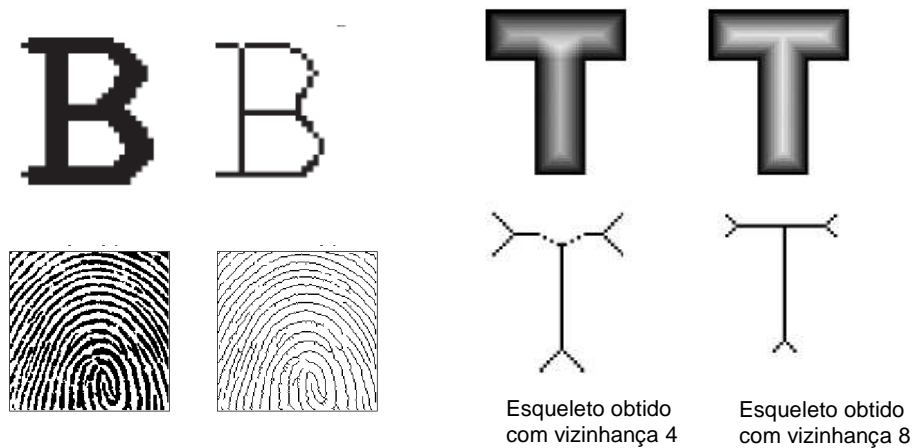
a

ere are only 10
people in the
who understand
and those who



O esqueleto representa a forma de um objeto por meio de um pequeno número de pixels representativos da sua posição, forma, orientação e comprimento.

Blum (1967) desenvolveu o algoritmo *medial axis function* (MAF) para extrair a linha de simetria (esqueleto) do objeto. Este algoritmo é sensível ao tipo de distância considerada (vizinhos 4, vizinhos 8, distância euclidiana,...).



SANTOS, Dalmar J. **Extração semi-automática do cruzamento de vias de áreas urbanas com uso de ortoimagens de alta resolução espacial.** Dissertação de mestrado. Curso de Pós-Graduação em Ciências Geodésicas, 2010.

Extração do esqueleto

- O esqueleto de A pode ser expresso através de erosões e aberturas:

$$S(A) = \bigcup_{k=0}^n S_k(A)$$

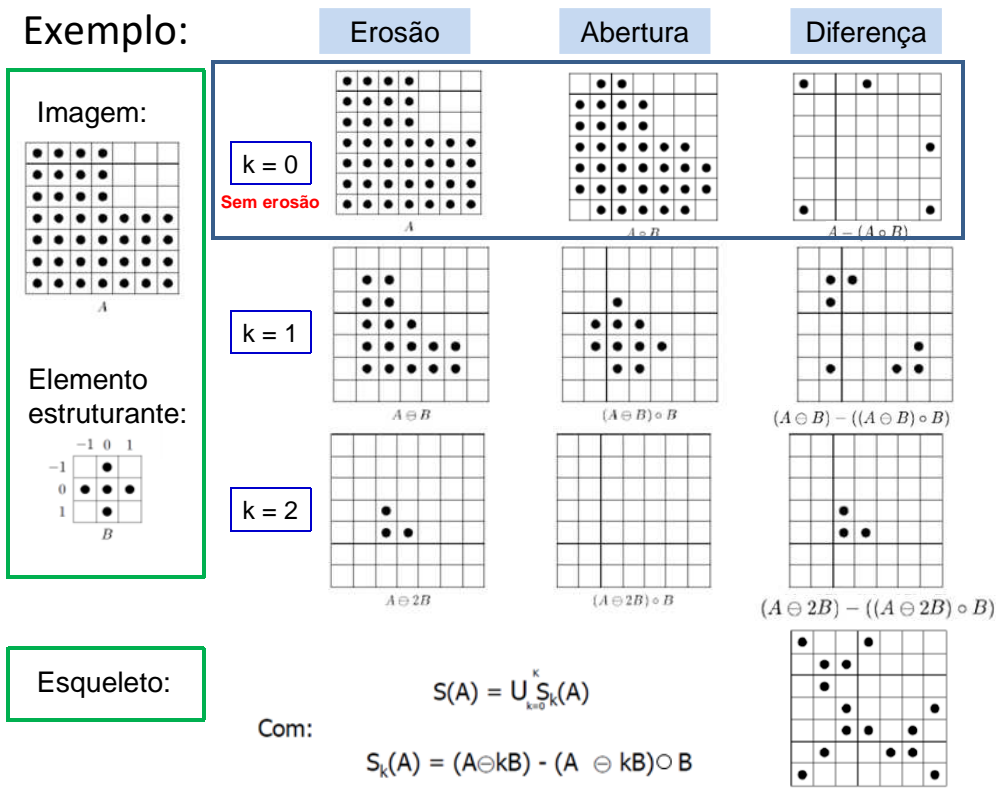
Com:

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

Em que :

- B é o elemento estruturante e
 - $(A \ominus kB)$ são k sucessivas erosões de A por B:
- $$(A \ominus kB) = (\dots ((A \ominus B) \ominus B) \ominus B)$$
- Repetido até que o último momento em que a próxima erosão resultará em um conjunto vazio.

Exemplo:



```

I = [
0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 0 0
0 1 1 1 1 0 0 0 0
0 1 1 1 1 0 0 0 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0]

```

```

EE = [
0 1 0
1 1 1
0 1 0
]

```

```

Ima = I;
it = 1

```

```

[m,n] = size(I);

```

```

% S: esqueleto
S = zeros(m,n);

```

```

[me,ne] = size(EE)

```

```

d1 = ceil(me/2);

```

```

d2 = floor(ne/2);

```

```

maximoS1 = 10 % depende do objeto e do elemento estruturante.

```

```

while maximoS1 > 0

```

```

    % Erosão

```

```

    S1 = zeros(m,n);

```

```

    ....

```

```

    % Abertura

```

```

    S2 = zeros(m,n);

```

```

    ....

```

```

    Aux = I - S2;

```

```

    % esqueleto:

```

```

    S = S + Aux;

```

```

I = S1;

```

```

    it = it + 1

```

```

    maximoS1 = max(max(S1))

```

```

end

```

$$(A \ominus kB)$$

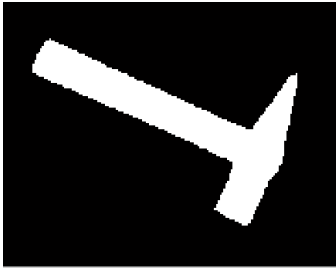
$$(A \ominus kB) \circ B$$

$$(A \ominus kB) - (A \ominus kB) \circ B$$

Esqueleto: $S(A) = \bigcup S_k(A)$

Exercício 1:

Dada a figura abaixo, encontrar o esqueleto do objeto mostrado abaixo utilizando o elemento estruturante EE:

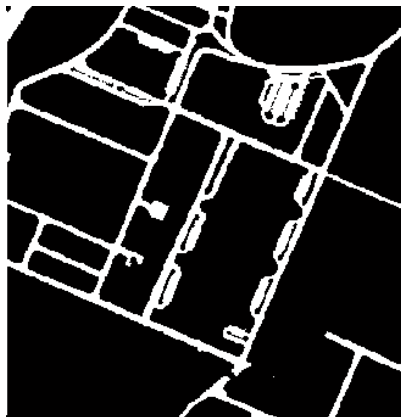


```
EE = [  
0 1 0  
1 1 1  
0 1 0  
]
```

Imagem: *ferramenta5.tif*

Exercício 2:

Aplique o operador esqueletização à imagem 'RuasCentroPolitecnico.tif'. Qual deve ser a forma do elemento estruturante EE para manter a conectividade?



Afinamento

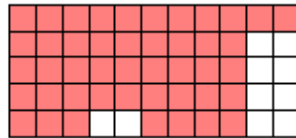
Serra (1982): Afinamento de um conjunto A por um elemento estruturante B:

$$A \otimes B = A - (A \otimes B)$$

onde

$$A \otimes B = (A \ominus B_1) \cap (A \ominus B_2) \quad \text{é o operador } \textit{hit or miss} \text{ (acerto e erro).}$$

A é a imagem :

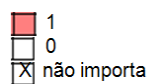
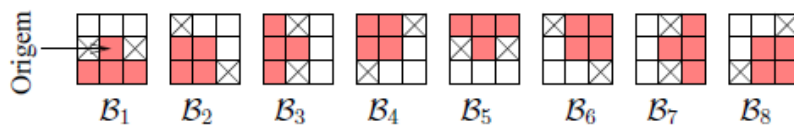


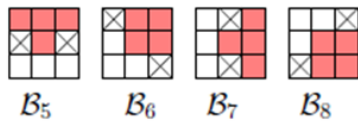
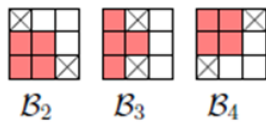
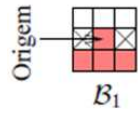
Afinamento

Outra implementação do afinamento de um conjunto A pelo elemento estruturante B:

$$A \circledast \{B\} = (((((A \otimes B_1) \otimes B_2) \otimes \dots) \otimes B_n)$$

Utiliza-se um conjunto B de elementos estruturantes $B_i = \{ B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8 \}$





%Elementos estruturantes

$B1 = [0\ 0\ 0; 0\ 1\ 0; 1\ 1\ 1]$

$B1C = [1\ 1\ 1; 0\ 0\ 0; 0\ 0\ 0]$

$B2 = [0\ 0\ 0; 1\ 1\ 0; 1\ 1\ 0]$

$B2C = [0\ 1\ 1; 0\ 0\ 1; 0\ 0\ 0]$

$B3 = [1\ 0\ 0; 1\ 1\ 0; 1\ 0\ 0]$

$B3C = [0\ 0\ 1; 0\ 0\ 1; 0\ 0\ 1]$

$B4 = [1\ 1\ 0; 1\ 1\ 0; 0\ 0\ 0]$

$B4C = [0\ 0\ 0; 0\ 0\ 1; 0\ 1\ 1]$

$B5 = [1\ 1\ 1; 0\ 1\ 0; 0\ 0\ 0]$

$B5C = [0\ 0\ 0; 0\ 0\ 0; 1\ 1\ 1]$

$B6 = [0\ 1\ 1; 0\ 1\ 1; 0\ 0\ 0]$

$B6C = [0\ 0\ 0; 1\ 0\ 0; 1\ 1\ 0]$

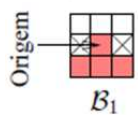
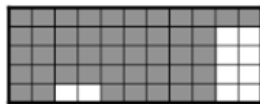
$B7 = [0\ 0\ 1; 0\ 1\ 1; 0\ 0\ 1]$

$B7C = [1\ 0\ 0; 1\ 0\ 0; 1\ 0\ 0]$

$B8 = [0\ 0\ 0; 0\ 1\ 1; 0\ 1\ 1]$

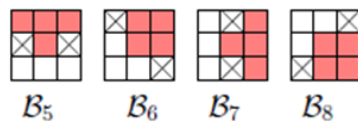
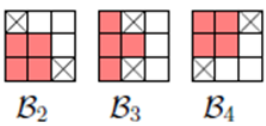
$B8C = [1\ 1\ 0; 1\ 0\ 0; 0\ 0\ 0]$

Imagem:

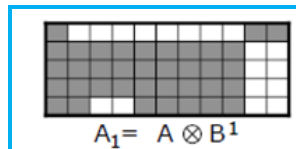


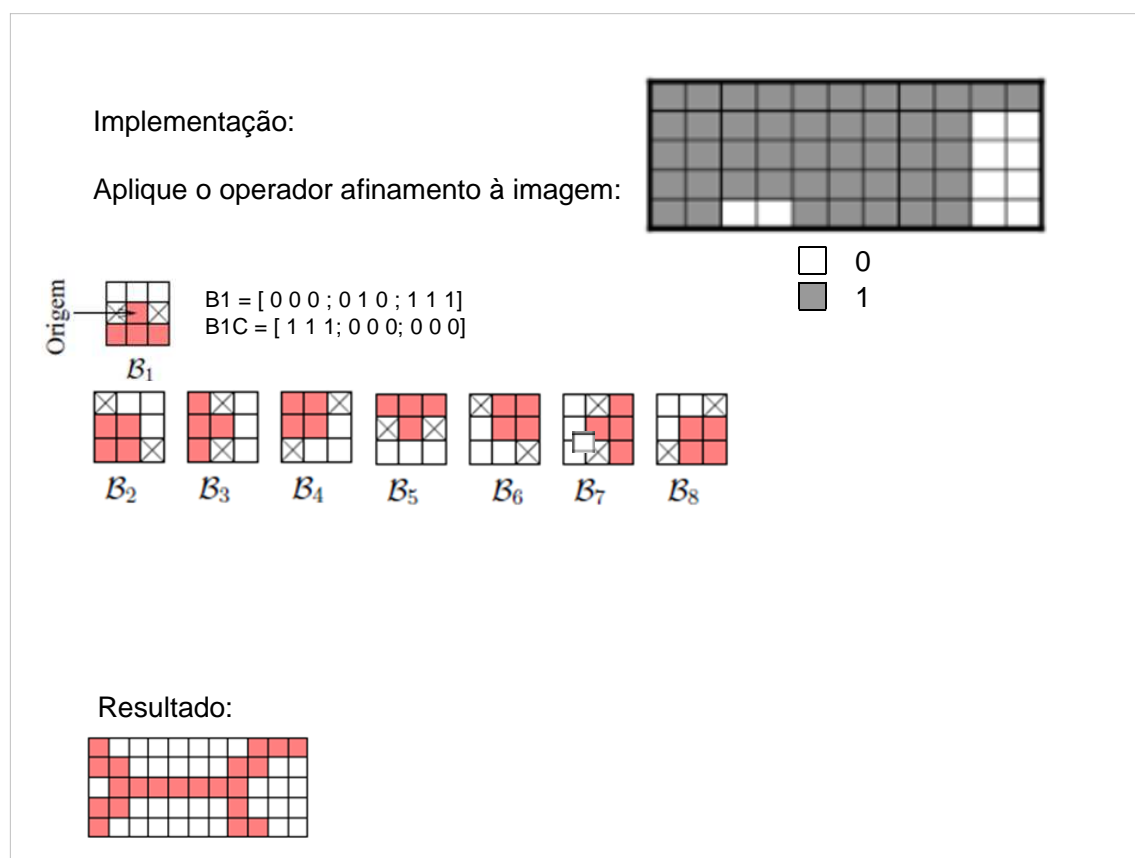
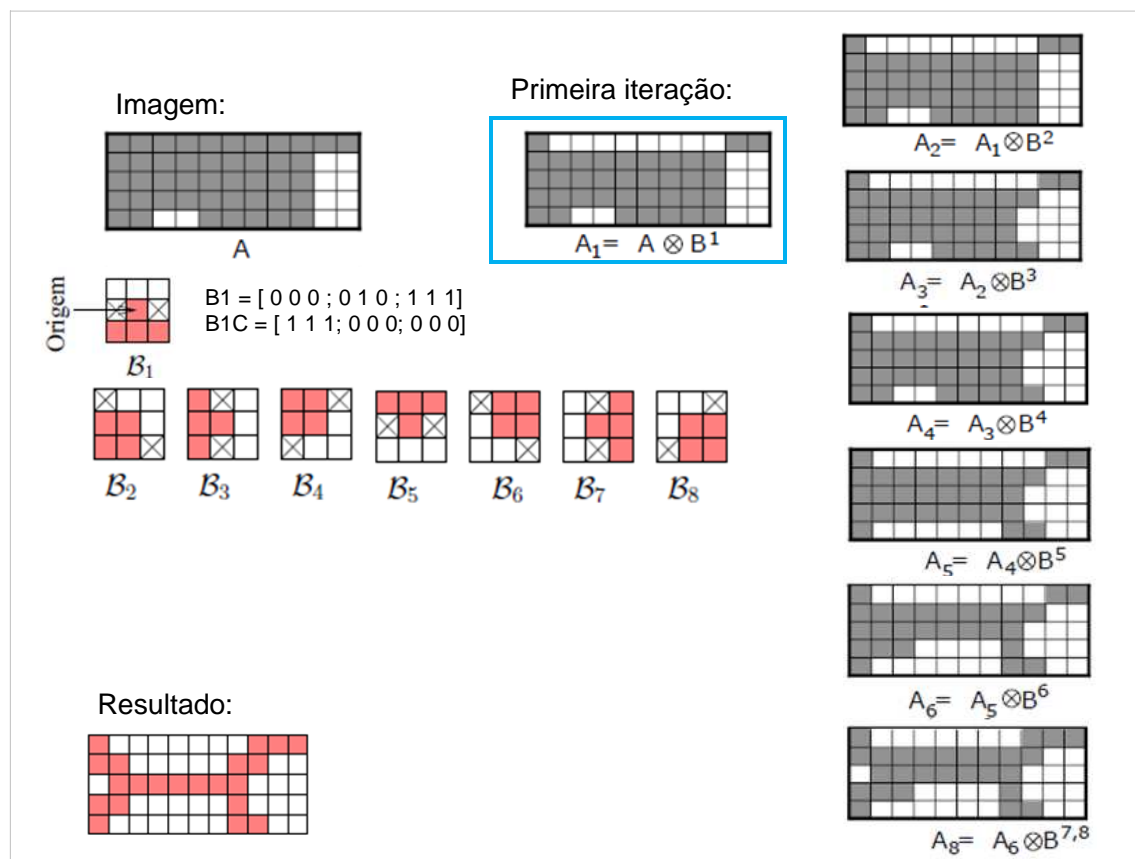
$B1 = [0\ 0\ 0; 0\ 1\ 0; 1\ 1\ 1]$

$B1C = [1\ 1\ 1; 0\ 0\ 0; 0\ 0\ 0]$



Primeira iteração:





```

[m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
it = 1
Filtro = 1
while it < 10

    while Filtro < 9
        % escolher filtros B11 e B11C
        % Bi = ...
        % BiC = ...

        % Aplicar filtros

        laux = I - S;
        I1(:, :, Filtro) = laux;
        I = laux;

        Filtro = Filtro + 1;
    end

    it = it + 1;
    Filtro = 1;

end

```

```

if Filtro == 1
    Bi = B1;
    BiC = B1C;
elseif Filtro == 2
    Bi = B2;
    BiC = B2C;
elseif Filtro == 3
    Bi = B3;
    BiC = B3C;
elseif Filtro == 4
    Bi = B4;
    BiC = B4C;
elseif Filtro == 5
    Bi = B5;
    BiC = B5C;
elseif Filtro == 6
    Bi = B6;
    BiC = B6C;
elseif Filtro == 7
    Bi = B7;
    BiC = B7C;
elseif Filtro == 8
    Bi = B8;
    BiC = B8C;
end

```

```

m,n] = size(I);
I1 = zeros(m,n,8);
S1 = zeros(m,n);
S2 = zeros(m,n);
S = zeros(m,n);
it = 1
Filtro = 1
while it < 18

    while Filtro < 9
        % escolher filtros B11 e B11C
        % Bi = ...
        % BiC = ...

        % Aplicar filtros

        laux = I - S;
        I1(:, :, Filtro) = laux;
        I = laux;

        Filtro = Filtro + 1;
    end

    it = it + 1;
    Filtro = 1;

end

```

```

for i = 2:m-1
    for j = 2:n-1
        ind = 0;
        for k = -1:1
            for l = -1:1
                if B11(2+k, 2+l) == 1 & I(i+k,j+l) == 1
                    ind = ind + 1;
                end
            end
        end
        if ind == 4
            S1(i,j) = I(i,j);
        else
            S1(i,j) = 0;
        end

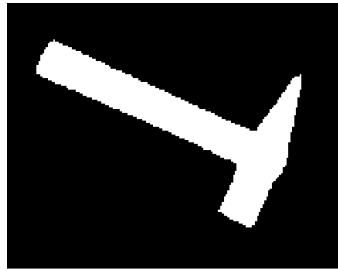
        ind = 0;
        for k = -1:1
            for l = -1:1
                if B11C(2+k, 2+l) == 1 & (1 - I(i+k,j+l)) == 1
                    ind = ind + 1;
                end
            end
        end
        if ind >= 3
            S2(i,j) = 0; % 1 - I(i+k,j+l) ;
        else
            S2(i,j) = 1;
        end
    end
end

for i = 2:m-1
    for j = 2:n-1
        if S1(i,j) == 1 & S2(i,j) == 0
            S(i,j) = 1;
        else
            S(i,j) = 0;
        end
    end
end
end

```

Exercício 3:

Aplique o operador afinamento à imagem 'ferramenta5.tif' considerando 4 iterações:



Exercício 4:

Como tornar a implementação do operador afinamento independente do número de iterações?

Aplique o programa modificado à imagem *ferramenta5.tif* e verifique quantas iterações foram necessárias para obter o esqueleto da figura.

